**IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE**

CYBERFONE SYSTEMS, LLC, Plaintiff, v. CELLCO PARTNERSHIP, ET AL., Defendants.	Civil Action No. 1:11-cv-827-SLR
CYBERFONE SYSTEMS, LLC, Plaintiff, v. AVAYA, INC., et al., Defendants.	Civil Action No. 1:11-cv-830-SLR
CYBERFONE SYSTEMS, LLC, Plaintiff, v. SONY ELECTRONICS, INC., et al., Defendants.	Civil Action No. 1:11-cv-833-SLR
CYBERFONE SYSTEMS, LLC, Plaintiff, v. FEDERAL EXPRESS CORP., et al., Defendants.	Civil Action No. 1:11-cv-834-SLR

PLAINTIFF CYBERFONE SYSTEMS, LLC'S ANSWERING BRIEF IN SUPPORT OF ITS EARLY CLAIM CONSTRUCTIONS

TABLE OF CONTENTS

PAGE

TABLE OF AUTHORITIES

I. NATURE AND STAGE OF PROCEEDINGS

Plaintiff Cyberfone Systems LLC ("Cyberfone") asserts one or more of U.S. Patent Nos. 5,805,676 ("the '676 Patent"), 5,987,103 ("the '103 Patent"), 6,044,382 ("the '382 Patent"), and 7,334,024 ("the '024 Patent") (the "Asserted Patents") against the remaining Defendants in the four remaining litigations.[1] At the January 31, 2013 status conference, the Court granted Defendants' request for early claim construction (*i.e.*, prior to plenary discovery and prior to construction of the rest of the disputed terms) on two of the disputed terms in the Asserted Patents ("form driven operating system" and "terminal assembly server (TAS)"). (D.I. 330.) The Court also granted Defendants' request to argue that two terms of the '024 Patent ("client module" and "computer program code for generating a data transaction") are synonymous with form driven operating system and TAS, respectively.

Cyberfone submits this brief in support of its proposed constructions.[2]

II. SUMMARY OF THE ARGUMENT

In their attempt to seek constructions that will "necessarily lead to summary judgment of noninfringement" (Def. Op. Br. at 1), Defendants ignore much of the intrinsic record, focus on one alleged *characteristic* of a form driven operating system (to the exclusion of addressing the actual functionality of a form driven operating system), and make no attempt at all to describe or define a terminal assembly server. Although Defendants espouse standard claim construction truisms throughout their Opening Brief, the actual constructions that Defendants offer are not faithful to the intrinsic record and run afoul of these principles.

[1] Given license agreements that have recently been finalized and signed, Cyberfone anticipates that fewer than twenty defendants total will remain across all four cases shortly.

[2] Defendants' claim in its Opening Brief that a "proper construction of these terms would necessarily lead to summary judgment of noninfringement" is false. Not only are Defendants' constructions improper, but factual disputes (and the need for fact discovery) will remain regardless of whether the Court adopts Cyberfone's constructions or Defendants' constructions.

Cyberfone's constructions, on the other hand, are not only faithful to the intrinsic record, but actually give meaning to the disputed terms. Indeed, Defendants' argument regarding one of the '024 terms ("computer program code for generating a data transaction") simply reinforces Cyberfone's construction of the term terminal assembly server. For all of the reasons below, Cyberfone requests that the Court adopt its proposed constructions of form driven operating system and terminal assembly server.

III. STATEMENT OF FACTS

A. Dr. Martino Identified A Problem With Expensive Telephone/Computer Systems.

In the early 1990s, Dr. Martino recognized that conventional operating systems at the time, as is still the case today, were "quite costly to purchase and maintain." ('676 Patent, col. 2:3-5.) As the specification for the '382 Patent notes, these conventional operating systems also typically "require[d] a great amount of memory and utilize[d] a great deal of processor overhead to operate efficiently." ('382 Patent, 2:40-42.) Thus, one of his goals was to create a data entry system "which does not have the inherent limitations of conventional point-of-entry systems such as the requirement of a **standard operating system** for communication with a remote service bureau or file server." ('676 Patent, 2:6-10 (emphasis added).)

B. The World Envisioned By Dr. Martino Came To Pass.

Dr. Martino's vision of a connected world in which devices such as mobile telephones, electronic gaming devices, media players, and mobile scanners would allow users to conduct transactions and leverage the power of remote computer servers – without the cost, memory, or processor burdens of a conventional operating system – proved prescient. The accused products in this litigation fall within the scope of these inventions. Instead of running full conventional operating systems, these devices employ software that is cheaper, less complex, and requires less memory and processing power to run.

IV. ARGUMENT

A. Legal Standards.

This Court is familiar with the principles of claim construction, so Cyberfone will not repeat them here. In light of Defendants' claim construction positions on the two disputed terms for which they would offer constructions, however, a couple principles deserve mention. First, claim construction is meant to determine what the applicant regards as his invention, not to craft definitions in relation to any particular products. 35 U.S.C. § 112(2); *Phillips v. AWH Corp.*, 415 F.3d 1303, 1311-12 (Fed. Cir. 2005) (en banc) (courts cannot "import limitations into claims from examples or embodiments appearing only in a patent's written description, even when a specification describes very specific embodiments of the invention or even describes only a single embodiment"). In other words, the proper construction of terms cannot be driven solely by a defendant's attempts to avoid infringement. Second, the construction of a term should clarify that term's meaning to a person of ordinary skill in the art. *See Phillips*, 415 F.3d at 1316-17. A construction that fails to give insight into a term's specific meaning within the context of the invention cannot be not the proper meaning.

B. Interaction Between "Form Driven Operating System" And "Terminal Assembly Server (TAS)."

Cyberfone and Defendants agree that a form driven operating system consists of a terminal assembly server (TAS) working in connection with forms:

> "As noted above, the **transaction assembly (application) server (TAS)** is a data stream stored in TAS PROM 95 which **together with the forms** from form/menu memory 96 **create a simple form driven operating system** which provides the necessary control data (firmware) to microprocessor 94 so that no conventional operating system is necessary." ('676 Patent, col. 16:11-16 (emphasis added).)

> "The intrinsic evidence is clear that the form driven operating system is a combination of the TAS firmware ***and*** forms that ***together*** provide the sole code to control the microprocessor." (Defendants' Opening Br. at 16 (emphasis in Def. Op. Br.).)
>
> "Like the specifications and prosecution histories of the Asserted Patents, Defendants' construction (i) defines the form driven operating system as a combination of firmware and forms …." (Defendants' Opening Br. at 14.)

Given the centrality of the TAS to both sides' understanding of a form driven operating system (in essence, that TAS plus forms equals form driven operating system), it would seem logical to define TAS. Yet only Cyberfone offers a construction for TAS that has any substance:

> the portion of a form driven operating system that performs at least the two basic functions of (1) generating a template or form from a data stream; and (2) developing a data transaction as the user inputs data in response to prompts in the template or form.

Defendants, on the other hand, offer but a nullity in their proposed construction of TAS, and simply refer back to a form driven operating system ("the firmware component of a form driven operating system"). As shown below, Defendants avoid the actual definition of TAS given in the intrinsic record because they would prefer that neither disputed term focus on the actual inventions by Dr. Martino. Instead, Defendants recast the term "form driven operating system" based on a single *characteristic* of a form driven operating system, and even there Defendants are not faithful to the actual language of the specification. Cyberfone has offered the only constructions of form driven operating system and TAS that provide substance for these terms. Cyberfone's constructions should be adopted, and Defendants' constructions should be rejected.

C. "Form Driven Operating System."

Cyberfone's Proposed Construction	Defendants' Proposed Construction
Computer code for developing data transactions, which, together with forms, controls the behavior of the microprocessor by logically defining a table of menu options and/or database interfaces. A form driven operating system is not a conventional operating system (such as DOS or Windows).	firmware together with forms that serve as the sole code for controlling a microprocessor, instead of an operating system that runs application programs

1. The Intrinsic Evidence Compels Cyberfone's Proposed Construction.

Cyberfone's proposed construction (unlike Defendants') actually defines "form driven operating system," and does so consistently with the specification. First, the specification makes clear that a form driven operating system is code for developing data transactions:[3]

- "The system of the invention provides for the automatic capture and computerization of data associated with ***data transactions*** as they occur." ('676 Patent, col. 5:50-52.)
- "entering ***data transactions*** into databases in accordance with the invention." (*Id.*, col. 7:9-11, 23-24.)
- "the creation and storage of a data transaction in accordance with the invention." (*Id.*, col. 9:15-17.)
- "the transaction entry device of the invention" (*Id.*, col. 10:52.)
- "[S]ince ***the data transactions*** are created without the use of an operating system or application programs, the transaction entry device is quite simple and inexpensive and may be readily integrated with the customer's desktop telephone or portable telephone." (*Id.*, col. 2:37-42 (emphasis added).)

The centrality of developing data transactions is reflected in the titles of the patents themselves: "Telephone/transaction entry device and system for entering ***transaction data*** into ***databases***" ('676 and '103 Patents) and "***Data transaction*** assembly server" ('382 Patent).[4] Yet despite the centrality of data transactions in the claimed inventions, Defendants' constructions do not even mention them. In other words, the very purpose and function of the inventions is absent from Defendants' constructions. As explained below, the reason for this is simple: Defendants wish to define these two disputed terms not in terms of function, but in terms of a skewed characteristic to forward their noninfringement position. Cyberfone's proposed construction of

[3] *See also* Def. Op. Br. at 3 ("All four Asserted Patents claim priority to the same 1995 application and disclose devices related to the same technical field: ***creating, transmitting, and storing data transactions***.") (footnote omitted and emphasis added).

[4] The '024 Patent, which does not have the same two disputed claim terms, is entitled "System for transmission of voice and data over the same communication line."

form driven operating system, as an initial matter, gives life to the function of the invention, which is intimately related to the generation and communication of data transactions.

Next, both sides agree that a form driven operating system controls a microprocessor (Cyberfone: "controls the behavior of a microprocessor"; Defendants: "controlling a microprocessor"). Cyberfone's construction, however, explains ***how*** this occurs: "by logically defining a table of menu options and/or database interfaces." This language also comes directly from the patent specification:

> "The TAS firmware and the selected template together control the behavior of the microprocessor 94 ***by logically defining a table of menu options and/or database interfaces*** which are navigated through by the user." '676 Patent, col. 14:5-8 (emphasis added).

This definition, again, gives meaning to the term and helps explain how a form driven operating system works and what makes an operating system "form driven." Defendants simply ignore the intrinsic record on this point.

The last part of Cyberfone's definition – that "a form driven operating system is not a conventional operating system (such as DOS or Windows)" – comes directly from the prosecution history. In his March 18, 1997 Response, the applicant explained that he amended the claims to recite a "form driven operating system" in order to overcome the examiner's belief that the prior version of the claims "would read on a general purpose computer having a microprocessor running a convention operating system (such as DOS or WINDOWS) and an application program . . . ." (*See* Murphy Decl., Ex. E at 18 ('676 Pat. File History, March 18, 1997 Resp.).)

Cyberfone's construction comes directly from the intrinsic record, the specification, and the file history, and gives meaning to the terms. Only Cyberfone's construction explains, for example, what makes a "form driven operating system" form driven, as opposed to conventional.

Finally, Defendants also strangely complain regarding Cyberfone's use of the phrase "together with forms":

> Although CyberFone's "computer code" controls the microprocessor "together with forms," the construction is contrary to the teachings of the specification because it treats the forms as being separate and distinct from the "form driven operating system." Indeed, by merely requiring that computer code be present "together with forms," it seeks to write "form driven" out of the claims.

(Def. Op. Br. at 16.) Defendants' complaint is strange because Defendants' own construction uses the same phrase, also defining form driven operating system in terms of TAS plus forms ("firmware together with forms"). Although Cyberfone disagrees with Defendants regarding the proper construction of TAS, Cyberfone agrees that TAS plus forms equals the form driven operating system. (*See, e.g.*, '676 Patent, col. 14:5-8 (emphasis added) ("The TAS firmware and the selected template ***together control*** the behavior of the microprocessor 94 by logically defining a table of menu options and/or database interfaces which are navigated through by the user."); *id.*, col. 13:50-53 (TAS and forms "operate together as a simple form driven operating system"); *id.*, col. 14:13-16 (TAS and menus and forms "together replace a conventional operating system and individual application programs").) Cyberfone's phrase "together with forms" was meant to signal that the computer code for generating data transactions, in combination with forms, ***together*** control the behavior of a microprocessor by logically defining a table of menu options and/or database interfaces. Although Cyberfone thought its use of Defendants' language was clear, Cyberfone is amenable to slight adjustments in the language of its proposed construction to avoid any doubt.

Finally, Defendants also contest Cyberfone's use of the term "computer code" in its proposed construction of form driven operating system. According to Defendants, "the specification consistently associates the TAS portion of the form driven operating system with descriptions such as "firmware," "microcode," and "PROM"—limitations not captured by

CyberFone's construction [of TAS]." (Def. Op. Br. at 15.) Given these varied descriptions of the TAS *within the specification itself*, it would be improper to read into the claims a limitation that TAS comprises solely firmware. The patentee never distinguished his inventions from the prior art by describing TAS as firmware (versus, for instance, software). Defendants can point to no clear disavowal by the patentee regarding the scope of the term TAS, and thus computer software code is the appropriate phrase for this part of the construction of the term form driven operating system. Moreover, as demonstrated below, Defendants explicitly argue that the TAS is also "computer program code for generating a data transaction." "Computer program code" is nearly identical to "computer software code," and stands in clear distinction to Defendants' attempt to read into the terms a limitation not present in the intrinsic record.

2. Defendants' Construction Is Incomplete And Ignores The Specification.

Defendants' proposed construction of form driven operating system is driven by its emphasis on one alleged characteristic of a form driven operating system, based on a phrase that appears *once* in the specification for the '676 patent:

> "The microcode of the TAS PROM 95 and the parameter streams from the form/menu memory 96 thus operate together as a simple form driven operating system for microprocessor 94 for all applications and is the sole code used to control microprocessor 94 (i.e., no conventional operating system or application programs are provided)." ('676 Patent, col. 13:50-56.)

Although the specification immediately explains what "sole code used to control microprocessor 94" means – *i.e.*, that no ***conventional*** operating system or application programs are provided – Defendants ignore this definition.[5] Instead, Defendants wish to use this one phrase (sole code) in

[5] "I.e." is Latin for "id est," or "that is [to say.]" Here, "i.e." is definitional. *See, e.g.*, *Abbott Labs v. Novopharm Ltd.*, 323 F.3d 1324, 1327, 1330 (Fed. Cir. 2003). To the extent that Defendants seek to undercut this understanding of the term "i.e." through *Dealertrack, Inc. v. Huber*, 674 F.3d 1315 (Fed. Cir. 2012), their argument fails. In *Dealertrack*, the court concluded that "the only way that the 'i.e.' in this patent could be read definitionally is if it

isolation, with the goal of subverting the rest of the specification and claims, and creating a hook to argue noninfringement. The actual sentence from the specification, however, does not permit such a misreading.

Defendants also misstate a key element of the one passage on which they do rely. The specification makes clear that a form driven operating system stands in contrast to a ***conventional*** operating system.

> "The microcode of the TAS PROM 95 and the parameter streams from the form/menu memory 96 thus operate together as a simple form driven operating system for microprocessor 94 for all applications and is the sole code used to control microprocessor 94 (i.e., no **conventional operating system** or application programs are provided)." ('676 Patent, col. 13:50-56.)
>
> "As noted above, the transaction assembly (application) server (TAS) is a data stream stored in TAS PROM 95 which together with the forms from form/menu memory 96 create a simple form driven operating system which provides the necessary control data (firmware) to microprocessor 94 so that no **conventional operating system** is necessary." ('676 Patent, col. 16:11-16 (emphasis added).)
>
> "As will be apparent from the following description, data transaction assembler 18 does not utilize a **conventional operating system** to control the operation of microprocessor 94." ('676 Patent, col. 13:38-41.)
>
> The objective of the invention is "[e]limination of the requirement of a **conventional operating system** and the associated application programs . . . ." ('676 Patent, 1:61-2:1 (emphasis added).)
>
> "TAS 18 does not utilize a **conventional operating system** to control the processing of application software." ('382 Patent, 16:43-45 (emphasis added).)

Defendants' proposed construction, however, omits the word "conventional," and attempts to rewrite the definition of form driven operating system to advance their noninfringement position. Defendants' proposal should be rejected.

excluded from the claim scope the embodiments discussed throughout the claim where only a single funding source is selected." *Dealertrack*, 674 F.3d at 1326. The *Dealertrack* court also noted that the term "i.e." was used throughout the patent to supply examples. *Id.* Neither of these situations is present here.

Finally, it must be noted that Defendants' construction, other than the phrase "together with forms," does not even mention the forms and menus at the heart of Dr. Martino's invention. In other words, the functionality of a form driven operating system is ignored, with the goal of focusing on a single phrase (sole code) to create a noninfringement argument. Because Cyberfone's construction is true to the intrinsic record and gives meaning to the term, and because Defendants' construction ignores key elements of the invention, misstates the record, and is phrased to work mischief in an infringement analysis, Cyberfone requests that the Court adopt Cyberfone's construction of form driven operating system and reject Defendants' construction.

3. "Client Module" Of '024 Patent.

Defendants admit that the term "client module" has a plain and ordinary meaning to one of ordinary skill in the art. (*See* Def. Op. Br. at 18 ("In the software context, a 'module' is defined as 'a program unit that is discrete and identifiable' or 'a logically separable part of a program.'") (quoting The IEEE Standard Dictionary of Electrical and Electronics Terms (5th ed.1993)).) Defendants also admit that "claim terms ordinarily take on "the meaning that a person of ordinary skill in the art would attribute to them . . . ." (*See* Def. Op. Br. at 11 (internal citations omitted).) Defendants point out that this plain and ordinary meaning may be supplanted when "(1) the patentee has chosen to be his own lexicographer, or (2) a claim term lacks such clarity that there is 'no meaning by which the scope of the claim may be ascertained from the language used.'" (*Id.* (citing *Novartis Pharma. Corp. v. Abbott Labs.*, 375 F.3d 1328, 1334 (Fed. Cir. 2004) (citing *Johnson Worldwide v. Zebco*, 175 F.3d 985, 990 (Fed. Cir. 1999)).) It is plainly obvious that neither situation applies here. The term "client module" is a term known to persons of ordinary skill in the art – Dr. Martino did not coin it. Nor does the term "lack[] such clarity that there is 'no meaning by which the scope of the claim may be ascertained.'" Avoiding these principles, Defendants turn to the "relatively narrow" additional interpretative principle of construing claims, if possible, to sustain their validity. *See generally* Patent Case Management Judicial Guide, Second Edition (2012) 5.2.3.4, at 5-87. Defendants, however, fail to acknowledge that this interpretative principle is only employed ***when a claim term is ambiguous***. The Federal Circuit has made this clear:

> "In this case, unlike in *Klein* and other cases in which the doctrine of construing claims to preserve their validity has been invoked, the claim term at issue is **not ambiguous**. Thus, it can be construed without the need to consider whether one possible construction would render the claim invalid while the other would not. The doctrine of construing claims to preserve their validity, a doctrine of limited utility in any event, therefore **has no applicability** here."

Phillips v. AWH Corp., 415 F.3d at 1328 (emphasis added). Defendants admit that the term "client module" is not ambiguous. This interpretive principle does not apply, and the term should be given its plain and ordinary meaning.

Defendants also, in passing, claim that applying the plain and ordinary meaning of "client module" in the claims of the '024 Patent would encompass matter that "patentee explicitly disclaimed," and provide a citation to the '676 Patent as support. (*See* Def. Op. Br. at 18 (citing '676 Patent, col. 2:58-60).) Cyberfone fails to see how the cited passage in any way supports Defendants' argument. Moreover, Defendants have not shown that applying the plain and ordinary meaning to the term "client module" would somehow expand the scope of the actual claims of the '024 patent to encompass matter that was explicitly disclaimed. The Court should reject Defendants' request to ascribe the construction of "form driven operating system" from the '676, '103, and '382 Patents to the term "client module" in the '024 Patent.

D. <u>"Transaction Assembly Server (TAS)."</u>

Cyberfone's Proposed Construction	Defendants' Proposed Construction
The portion of a form driven operating system that performs at least the two basic functions of (1) generating a template or form from a data stream; and (2) developing a data transaction as the user inputs data in response to prompts in the template or form.	the firmware component of a form driven operating system

1. The Intrinsic Evidence Compels Cyberfone's Construction.

As noted above, both sides agree that the terminal assembly server or TAS, in combination with forms, together create a form driven operating system. Cyberfone's construction of TAS comes directly from the specification:

> The TAS "in general performs the two basic functions of (1) generating a template or form from a data stream and (2) developing a data transaction as the user inputs data in response to prompts in the template or form." ('676 Patent, col. 2:55-60.)

Defendants do not appear to seriously dispute that Cyberfone's construction accurately describes the functions of the TAS. Indeed, in the portion of Defendants' Opening Brief discussing the term "computer code for generating a data transaction" – which Defendants contend is equivalent to TAS – Defendants quote the specification at length, after noting that the TAS is the code used to generate a data transaction:

- The TAS "in general performs the two basic functions of (1) generating a template or form from a data stream and (2) ***developing a data transaction*** as the user inputs data in response to prompts in the template or form." ('676 Patent, col. 2:55-60.)

- The TAS firmware of the invention stores the options as well as control programs (microcode) for the processor for use with the templates in ***creating the data transactions***. ('676 Pat., col. 6:43-45.)

- Thus, the TAS PROM 95 contains control data (firmware) for the microprocessor 94 and resides in each transaction entry device 12 for ***generating a template for a data***

> ***transaction from a data stream stored in form/menu memory***. ('676 Pat., col. 13:65-14:1.)

(Def. Op. Br. at 22 (emphasis added by Defendants).) Each of these passages supports Cyberfone's construction.

Defendants, however, object that Cyberfone's construction of TAS contains "superfluous" limitations because, it argues, some of the characteristics of the TAS that Cyberfone delineates are also separately spelled out in claim 16 of the '676 Patent. For support, Defendants misapply *Unique Concepts, Inc. v. Brown*, 939 F.2d 1558 (Fed. Cir. 1991). In *Unique Concepts*, the court faced the construction within a single independent claim that made "unambiguous reference to **two distinct elements** of the claimed structure: linear border pieces and right angle corner pieces." *Id.* at 1561-62 (emphasis added). The Court noted: "If, as Unique argues, linear border pieces of framing material, whose ends are mitered, are the same as linear border pieces and a right angle corner piece, the **recitation of both types of pieces is redundant.**" *Id.* (emphasis added). This is clearly not the case with the asserted patents. And *Unique Concepts* contains many other distinguishing aspects, such as an agreement by the parties (criticized in the dissent) that the construction of the terms could not invalidate the patent.[6] In short, Defendants' complaint regarding Cyberfone's construction of TAS has no basis in law.

2. Defendants' Construction Does Nothing To Define TAS, Ignores The Specification, And Is Inconsistent With Defendants' Position On The '024 Patent.

In contrast to Cyberfone's definition of terminal assembly server, Defendants offer merely a nullity: a reference back to its (incomplete and inaccurate) definition of form driven operating system. Yet in arguing that the term "computer program code for generating a data

[6] *See also id.* at 1567-68 (Rich, Circuit Judge, dissenting) ("I am not "merging the two types of claim elements into one" — whatever that may mean.").

transaction," which appears in the '024 Patent, should be interpreted identically with TAS, Defendants cite the '676 specification repeatedly:

- The TAS "in general performs the two basic functions of (1) generating a template or form from a data stream and (2) ***developing a data transaction*** as the user inputs data in response to prompts in the template or form." ('676 Patent, col. 2:55-60.)
- The TAS firmware of the invention stores the options as well as control programs (microcode) for the processor for use with the templates in ***creating the data transactions***. ('676 Pat., col. 6:43-45.)
- Thus, the TAS PROM 95 contains control data (firmware) for the microprocessor 94 and resides in each transaction entry device 12 for ***generating a template for a data transaction from a data stream stored in form/menu memory***. ('676 Pat., col. 13:65-14:1.)

(Def. Op. Br. at 22 (emphasis added by Defendants).) Each of these passages supports ***Cyberfone's*** construction of TAS, which gives actual meaning and definition to the term. Moreover, this formulation does ***not*** limit TAS to firmware – the claim language itself is "***computer program code***." This puts to rest Defendants' attempt to read into the definition of TAS a limitation not found in the claims, the specification, or the file history (that TAS must only be firmware).

3. "Computer Program Code For Generating A Data Transaction."

In arguing that the term "computer program code for generating a data transaction" should be given the same construction as TAS, Defendants explicitly argue that this computer code "is the code that is used to generate a data transaction," which is the same function as the TAS. (*See* Def. Op. Br. at 22; *see also id.* ("The identical function of these elements argues in favor of construing the terms to be synonymous.").) As noted above, Defendants point to the following passages from the '676 to show that the terms are identical:

- The TAS "in general performs the two basic functions of (1) generating a template or form from a data stream and (2) ***developing a data transaction*** as the user inputs data in response to prompts in the template or form." ('676 Patent, col. 2:55-60.)

- The TAS firmware of the invention stores the options as well as control programs (microcode) for the processor for use with the templates in ***creating the data transactions***. (‘676 Pat., col. 6:43-45.)
- Thus, the TAS PROM 95 contains control data (firmware) for the microprocessor 94 and resides in each transaction entry device 12 for ***generating a template for a data transaction from a data stream stored in form/menu memory***. (‘676 Pat., col. 13:65-14:1.)

Defendants’ Br. at 22 (emphasis added by Defendants). In doing so, Defendants effectively concede that both TAS and the “computer program code for generating a data transaction” match ***Cyberfone’s*** proposed construction of TAS. Moreover, this formulation clearly does not limit TAS to firmware – the claim language itself is “***computer program code***.” Cyberfone agrees that its construction of TAS is proper, and that the term “computer program code for generating a data transaction” should be construed to match Cyberfone’s construction of TAS.

V. CONCLUSION

For all of the reasons above, the Court should adopt Cyberfone’s proposed constructions of form driven operating system and terminal assembly server. The Court should also adopt Cyberfone’s proposed construction of terminal assembly server for the term “computer program code for generating a data transaction” in the ‘024 Patent. Finally, the Court should reject Defendants’ request that the term “client module” be found synonymous with form driven operating system.

Dated: April 3, 2013

Of Counsel:

Marc A. Fenster
Eric J. Carsten
Russ, August & Kabat
12424 Wilshire Boulevard, 12th Floor
Los Angeles, CA 90025-1031
(310) 826-7474
mfenster@raklaw.com
ecarsten@raklaw.com

BAYARD, P.A.

/s/ *Stephen B. Brauerman*
Richard D. Kirk (rk0922)
Stephen B. Brauerman (sb4952)
Vanessa R. Tiradentes (vt5398)
222 Delaware Avenue, Suite 900
Wilmington, DE 19801
(302) 655-5000
rkirk@bayardlaw.com
sbrauerman@bayardlaw.com
vtiradentes@bayardlaw.com

Attorneys for Plaintiff,
Cyberfone Systems, LLC